Readme

**1. 程序功能说明**

本程序能够实现模拟出租车的乘客呼叫与应答系统。与上次相比增加了可追踪出租车功能。

**2. 程序运行所需环境**

编写时使用了Eclipse。

**3. 输入规范**

3.1 在指导书的基础上做输入规范。

3.2 读入的地图文件路径默认为D://map.txt，如需更改请到Uber.java第95行更改路径，注意每级之间只能用“//”连接。

如果地图文件每个点之间没有分隔符，则不做修改。如果用空格或制表符分隔，则需要在Map.java第113行在括号中的引号中间添加空格或\t。

3.3 【新增】读入的红绿灯文件路径默认为D://light.txt，如需更改请到Uber.java第96行更改路径，注意每级之间只能用“//”连接。

如果红绿灯文件每个点之间没有分隔符，则不做修改。如果用空格或制表符分隔，则需要在Light.java第92行在括号中的引号中间添加空格或\t。

**注：所有红绿灯的间隔时间相同，即所有红绿灯南北方向同时为绿灯或东西方向同时为绿灯。GUI中，在等待红绿灯变绿时，出租车在原地停止运动，但是车的颜色不发生变化。**

3.4 订单在控制台输入，输入格式为“[CR,(srcX, srcY),(dstX, dstY)]”，支持中间空格，支持前导零，支持数字前一个“+”号。全部订单（包括无效订单）不得超过1000行。

3.5 一行可输入不超过10个有效订单，用“;”分割。

3.6 可以从控制台输入指定出租车的序号（0~99）来查询该车的状态，输入的数值包括前导零不得超过两位数，否则算作非法输入。不支持数字前“+”号和空格。

3.7 可以从控制台输入指定状态来查询处于指定状态的出租车对象，输入格式为“StatusX”，X的值**为1表示服务状态，为3表示接单状态，为2表示等待服务，为0表示停止运行**。除这四种外均为非法输入。

3.8可以从控制台输入指定请求来打开和关闭道路，输入格式为“[OPEN/CLOSE,(X1,Y1),(X2,Y2)]”，(X1,Y1)、(X2,Y2)为道路两端的点（顺序随意），**不支持空格**，支持前导零，支持数字前一个“+”号。输入的坐标值包括前导零不得超过两位数，否则算作非法输入。一次只能输入一个请求（一个OPEN或一个CLOSE）。

3.9 【新增】从Uber.java的第39行开始有一个initTaxi方法，测试者可通过修改该方法的代码来设置出租车类别，将编号为i的出租车设置为可追踪出租车的方法为a[i]=1；**可追踪出租车应恰好有30辆。**（如果不做修改，则默认0~99号出租车中30-59号为可追踪出租车，其余为普通出租车）

3.10 【新增】可以从控制台输入指定请求来查询可追踪出租车的乘客服务情况，正序的输入格式为“InfoX”，倒序的输入格式为“ReverseX”，X的值为0~99的一个整数。输入的数值包括前导零不得超过两位数，否则算作非法输入。不支持数字前“+”号和空格。**应当在出租车已达到目标点之后再输入该请求，否则输出的路径可能不完整。**

1. **输出规范**

4.1 如果目标路径地图文件不存在，控制台将输出“map.txt does not exist!”并直接退出。如果读入的地图文件不符合要求（每行80个数，数只能为0、1、2、3之一），控制台将输出“Invalid map!”并终止程序。

如果目标路径红绿灯文件不存在，控制台将输出“light.txt does not exist!”并直接退出。如果读入的地图文件不符合要求（每行80个数，数只能为0或1），控制台将输出“Invalid Traffic Lights!”并终止程序。

4.2 程序开始运行后，控制台出现“Ready for order!”（同时出现Gui界面）后才可以输入订单。

4.3 如果输入超过1000行指令，控制台将输出“There are more than 1000 lines of order.”并不再处理之后的输入。

4.4 如果输入不符合规范，控制台将输出“Invalid input: [输入内容]”；

如果输入能被判定为订单但是超出了地图范围，控制台将输出“Invalid order: [输入内容]”；

如果一行输入了超过10个有效订单，则对于10个之后的每个订单，控制台将输出“Invalid order(more than 10 orders in a line): [输入内容]”；

如果订单中的请求地点与目的地一样，控制台将输出“Invalid order(src=dst): [输入内容]”且不对该请求进行处理；

如果同一时刻在同一地点发出的去同一目的地的请求，视为相同请求，控制台将输出“Same order: [输入内容]”并忽略后一条请求。

4.5 当输入指定出租车的序号时，控制台的输出格式为：T: No.n Type :t at (x,y) status:s credit:c 其中**T为查询时间（单位ms），t为车的种类（0为普通出租车，1为可追踪出租车）n为车的序号，(x,y)为车当前的位置坐标，s为状态（1表示服务状态，3表示接单状态，2表示等待服务，0表示停止运行），c为信用度**。

4.6 当输入指定状态时，控制台会将符合状态的出租车序号在一行内输出，用空格分开；如果没有符合条件的就不输出。

4.7 当输入开关道路的指令时，

如果能被判定为开关路指令但是超出了地图范围，控制台将输出“Invalid request: [输入内容]”。

如果输入的两个端点相同或者不是某条边的端点，控制台将输出“Invalid request(not a road): [输入内容]”；

如果输入的是OPEN指令，而两个端点构成的边已经打开，控制台将输出“The road is already opened!”；如果输入的是CLOSE指令，而两个端点构成的边已经打开，控制台将输出“The road is already closed!”

如果关闭道路使得点不再是交叉路口或丁字路口，该点上的红绿灯保留，出租车仍然根据红绿灯的方向来运动。

4.8 对于到文件的输出规定如下：

4.8.1每产生一个请求，都在D://testtaxi目录下新建一个命名为“T(srcX,srcY)-(dstX,dstY).txt”的文件。如果目录不存在将自动创建；文件需手动删除。(...)

4.8.2 输出请求发出时，处于以请求src为中心的4×4区域中的所有出租车状态、信用信息的格式为：（此时的信用度为接单之前的信用度）

**Taxi No.n: at (x,y) status:s credit:c**

4.8.3 输出在抢单时间窗内所有抢单的出租车的格式为：

**Taxi in compete: No.n at (x,y)**

4.8.4 输出系统选择响应相应请求的出租车的格式为

**Choose: No.n**

4.8.5 输出出租车响应相应请求过程中的实际行驶路径的格式为

**(x,y)->...->(srcX,srcY)** ←从车当前坐标到订单发出点的最短路径

**(srcX,srcY)>...->(dstX,dstY)** ←从订单发出点到订单目标点的最短路径

4.8.6 如果没有出租车相应该请求，则输出

**No taxi answer the order:(**

注：各数值定义同4.5

4.9【新增】如果测试者设置的可追踪出租车的数目不是30，将输出“The number of traceable taxi is not 30!”，但其余功能照常运行。

4.10【新增】当输入可追踪出租车的乘客服务情况时，

如果输入序号对应的出租车不是可追踪出租车，控制台将输出“Taxi No.X is not a traceable taxi!”，X为出租车序号；

如果输入“InfoX”，则按照时间从小到大的顺序输出该出租车所抢到的乘客请求（请求产生时刻（单位ms）、请求发出位置->目的地位置），出租车在抢到单时的所处位置坐标，出租车去接乘客以及运送乘客去目的地途中所行驶的路径（路径格式同4.8.5）；

如果输入“ReverseX”，则按照时间从大到小的顺序输出上述信息；

上述两种情况，如果相应出租车没有接过单，则不输出。

1. **Gui测试说明**

5.1 出租车为服务状态时，显示为蓝色；为接单状态时，显示为绿色；为等待服务状态时，显示为红色；为停止运行状态时，显示为黄色。出租车到达请求发出地点和到达目的地后停止服务1s，此时Gui中车只会停止运行而不会改变颜色。【新增】可追踪出租车无论何种状态，始终显示为粉色。

**注：可追踪出租车能在开始运行时打开、运行过程中关闭的道路上行驶，但是不能在开始运行时就关闭的道路上行驶。**

5.2 为避免订单数量很多时Gui界面出现卡顿现象，将Order.java中第151行代码注释掉。如果测试者希望通过Gui界面查看订单发出地点是否正确，可将注释取消。

注：因为时间有限，方法规格和类规格采用了中英文结合的方式。

求大佬轻扣qwq